시스템 프로그래밍

Assignment1-3

Class : A

Professor : 김태석 교수님

Student ID : 2022202065

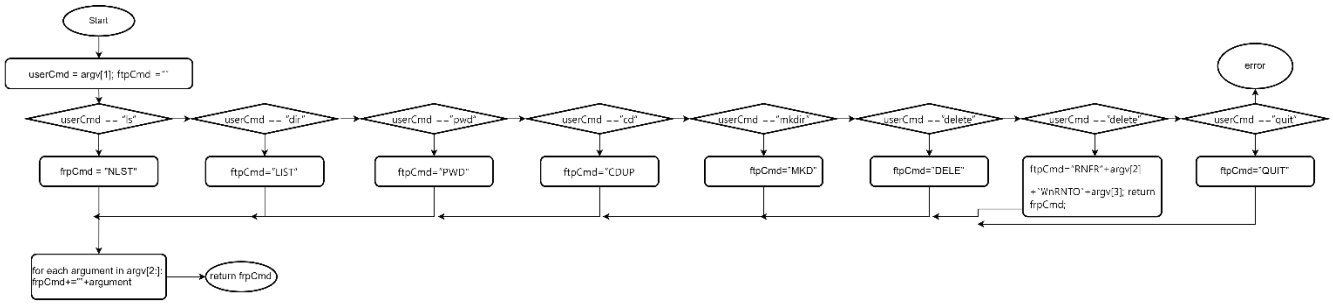
Name : 박나림

Introduction

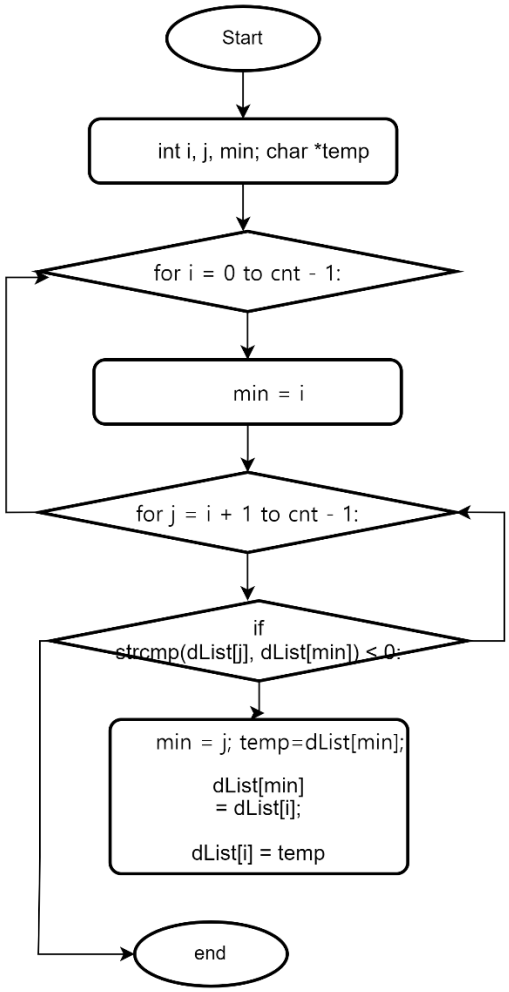
FTP (File Transfer Protocol) 프로젝트의 세번째 단계로, 리눅스의 user 명령어들을 FTP 명령어로 변환시키고 실행하는 것이 목표이다. 이 과정에서 client 와 server 가 통신하는 방식을 공부해본다. client 와 server 프로그램 두 개를 만들고, client 에서는 user 명령어들을 FTP 명령어로 변환시킨뒤 표준 출력으로 write 한다. 그러면 server 에서 read 로 받아서 다시 user 명령어로 변환시킨다. 그 후 각 명령어마다 함수를 구현하여 실행시키는 것으로 한다. 명령어는 ls (-a, -l, -al), dir, pwd, cd, mkdir, delete, rmdir, rename, quit 를 구현하도록 한다. 각 명령어 실행 중에 발생한 오류에 대해서도 그에 맞는 예외처리를 해야 한다. 최종적으로 FTP 명령어가 실행되는 과정을 구현하도록 한다.

Flow chart

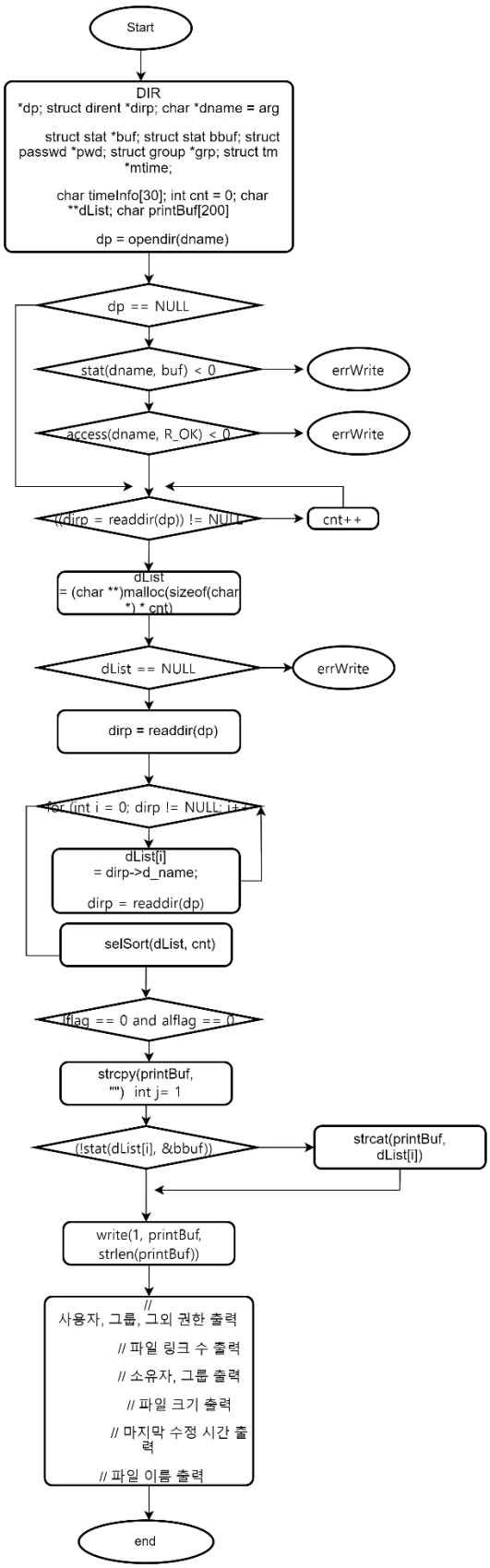
cli.c



srv.c – selSort



srv.c – lscmd


Start
DIR
*dp; struct dirent *dirp; char *dname = arg
struct stat *buf; struct stat bbuf; struct passwd *pwd; struct group *grp; struct tm *mtime;
char timeInfo[30]; int cnt = 0; char **dList; char printBuf[200]
dp = opendir(dname)
dp == NULL
stat(dname, buf) < 0
errWrite
access(dname, R_OK) < 0
errWrite
((dirp = readdir(dp)) != NULL)
cnt++
dList = (char **)malloc(sizeof(char *) * cnt)
dList == NULL
errWrite
dirp = readdir(dp)
for (int i = 0; dirp != NULL; i++)
dList[i] = dirp->d_name;
dirp = readdir(dp)
selSort(dList, cnt)
lflag == 0 and alflag == 0
strcpy(printBuf, "") int j= 1
(!stat(dList[i], &bbuf))
strcat(printBuf, dList[i])
write(1, printBuf, strlen(printBuf))
// 사용자, 그룹, 그외 권한 출력
// 파일 링크 수 출력
// 소유자, 그룹 출력
// 파일 크기 출력
// 마지막 수정 시간 출력
// 파일 이름 출력
end

Pseudo code

cli.c

```
char* converFTP(int argc, char* argv[]) {

userCmd = argv[1]

ftpCmd =""


if(userCmd == "ls") {
        ftpCmd = "NLST"
        for each option in argv:
                if option == "-a": aflag++
                else if option =="-l": lflag++
        if (aflag!=0 and lfag==0) ftpCmd += "-a"
        else if (aflag==0 and lflag !=0) ftpCmd+="-l"
        else {
                for each argument in argv:
                        ftpCmd+=argument
                        return ftpCmd
        }
        else if (userCmd =="dir") ftpCmd="LIST"
        else if (userCmd =="pwd") ftpCmd="PWD"
        else if (userCmd =="cd") {
                if (argv[2]="..") ftpCmd="CDUP
                else ftpCmd ="CMD"
                if (argv[2]==NULL) errArg()
        }
```

```
        else if (userCmd ==“mkdir”){
                ftpCmd=“MKD”
                if(argv[2]==NULL) errArg()
        }
        else if (userCmd ==“delete”) ftpCmd=”DELE”
                frpCmd=“DELE”
                if(argv[2]==NULL) errArg()
        }
        else if(userCmd==”rename”) {
                if(argv[2]==NULL) errArg()
                else { ftpCmd=“RNFR”+argv[2]+“₩nRNTO”+argv[3]; return frpCmd;}
        else if (userCmd ==“quit”) ftpCmd=“QUIT”
        else error
        for each argument in argv[2:]: frpCmd+=””+argument
        return frpCmd
}
```

srv.c

```
selSort(char *dList[], int cnt) {
    int i, j, min
    char *temp
    for i = 0 to cnt - 1:
        min = i
        for j = i + 1 to cnt - 1:
            if strcmp(dList[j], dList[min]) < 0:
                min = j
        temp = dList[min]
```

```
            dList[min] = dList[i]
            dList[i] = temp
}

lsCmd(int aflag, int lflag, int alflag, char *arg) {
        DIR *dp; struct dirent *dirp; char *dname = arg
        struct stat *buf; struct stat bbuf; struct passwd *pwd; struct group *grp; struct tm *mtime;
        char timeInfo[30]; int cnt = 0; char **dList; char printBuf[200]
        dp = opendir(dname)
        if (dp == NULL) {
            if (stat(dname, buf) < 0) {
                errWrite()
            } else if (access(dname, R_OK) < 0) {
                errWrite()
            }
        }
        while ((dirp = readdir(dp)) != NULL) {
            cnt++
        }
        dList = (char **)malloc(sizeof(char *) * cnt)
        if (dList == NULL) {
            errWrite()
        }
        dirp = readdir(dp)
        for (int i = 0; dirp != NULL; i++) {
            dList[i] = dirp->d_name
```

```
        dirp = readdir(dp)
    }
    selSort(dList, cnt)
    if (lflag == 0 and alflag == 0) {
        strcpy(printBuf, "")
        int j = 1
        for (int i = 0; i < cnt; i++) {
            if (aflag == 0 and dList[i][0] == '.') {
                continue
            }
            j++
            if (!stat(dList[i], &bbuf)) {
                strcat(printBuf, dList[i])
                if (S_ISDIR(bbuf.st_mode) == 1) {
                    strcat(printBuf, "/₩t")
                } else {
                    strcat(printBuf, "₩t")
                }
            }
            if (j % 5 == 0) {
                strcat(printBuf, "₩n")
            }
        }
        strcat(printBuf, "₩n")
        write(1, printBuf, strlen(printBuf))
    } else {
```

```
        for (int i = 0; i < cnt; i++) {
            if (lflag == 1 and dList[i][0] == '.') {
                continue
            }
            if (!stat(dList[i], &bbuf)) {
            if (S_ISREG(bbuf.st_mode) == 1) {
                    strcpy(printBuf, "-")
                } else {
                    strcpy(printBuf, "d")
                }
                // 사용자, 그룹, 그외 권한 출력
                // 파일 링크 수 출력
                // 소유자, 그룹 출력
                // 파일 크기 출력
                // 마지막 수정 시간 출력
                // 파일 이름 출력
                if (S_ISDIR(bbuf.st_mode) == 1) {
                    strcat(printBuf, "/₩n")
                } else {
                    strcat(printBuf, "₩n")
                }
            }
            write(1, printBuf, strlen(printBuf))
        }
    }
    if (closedir(dp) < 0) {
```

```
            errWrite("closed error₩n")
        } free(dList)
}
//parse 는 cli 와 동일한 알고리즘 이용
```

결과화면

1. NLST (ls)

1) ls 에 다른 옵션을 입력했을 경우, 예외처리 된다.

```
kw2022202065@ubuntu:~/1-3$ ./cli ls -e | tee cli.out | ./srv
./cli: invalid option -- 'e'
Error: invalid option
```

2) 기본 ls 를 했을 때는 숨김 파일이 보이지 않고, -a 까지 입력하면 숨김 파일이 표시된다. 디렉토리는 뒤에 '/'가 붙어서 출력된다. 한 줄에 4, 5 개씩 출력되도록 하였다. 출력 값은 오름차순으로 정렬된다.

```
kw2022202065@ubuntu:~/1-3$ ./cli ls | tee cli.out | ./srv
NLST
AA      Makefile        cle.out cli
cli.c   cli.out srv     srv.c   test

kw2022202065@ubuntu:~/1-3$ ./cli ls -a | tee cli.out | ./srv
NLST -a
./      ../     AA      Makefile
cle.out cli     cli.c   cli.out srv
srv.c   test
```

3) -l 과 -al 을 했을 때는 한 줄씩 자세하게 출력된다. 권한, 링크 수, 사용자 이름, 그룹 이름, 사이즈, 마지막 수정 날짜, 파일 및 디렉토리 이름이 출력된다.

```
kw2022202065@ubuntu:~/1-3$ ./cli ls -l | tee cli.out | ./srv
NLST -l
-rw-rw-r--      1       kw2022202065    kw2022202065    0       04 12 22:28     AA
-rw-rw-r--      1       kw2022202065    kw2022202065    72      04 11 20:06     Makefile
-rw-rw-r--      1       kw2022202065    kw2022202065    10      04 12 22:50     cle.out
-rwxrwxr-x      1       kw2022202065    kw2022202065    17200   04 12 21:25     cli
-rw-rw-r--      1       kw2022202065    kw2022202065    2240    04 12 23:09     cli.c
-rw-rw-r--      1       kw2022202065    kw2022202065    8       04 12 23:11     cli.out
-rwxrwxr-x      1       kw2022202065    kw2022202065    22312   04 12 23:08     srv
-rw-rw-r--      1       kw2022202065    kw2022202065    8344    04 12 23:10     srv.c
-rw-rw-r--      1       kw2022202065    kw2022202065    9       04 11 21:01     test
kw2022202065@ubuntu:~/1-3$ ./cli ls -al | tee cli.out | ./srv
NLST -al
drwxrwxr-x      2       kw2022202065    kw2022202065    4096    04 12 23:10     ./
drwxr-xr-x      22      kw2022202065    kw2022202065    4096    04 12 23:10     ../
-rw-rw-r--      1       kw2022202065    kw2022202065    0       04 12 22:28     AA
-rw-rw-r--      1       kw2022202065    kw2022202065    72      04 11 20:06     Makefile
-rw-rw-r--      1       kw2022202065    kw2022202065    10      04 12 22:50     cle.out
-rwxrwxr-x      1       kw2022202065    kw2022202065    17200   04 12 21:25     cli
-rw-rw-r--      1       kw2022202065    kw2022202065    2240    04 12 23:09     cli.c
-rw-rw-r--      1       kw2022202065    kw2022202065    9       04 12 23:12     cli.out
-rwxrwxr-x      1       kw2022202065    kw2022202065    22312   04 12 23:08     srv
-rw-rw-r--      1       kw2022202065    kw2022202065    8344    04 12 23:10     srv.c
-rw-rw-r--      1       kw2022202065    kw2022202065    9       04 11 21:01     test
```

2. LIST (dir)

1) ls -al 과 동일하게 출력된다.

```
kw2022202065@ubuntu:~/1-3$ ./cli dir | tee cli.out | ./srv
LIST
drwxrwxr-x      4       kw2022202065    kw2022202065    4096    04 12 23:21     ./
drwxr-xr-x      22      kw2022202065    kw2022202065    4096    04 12 23:21     ../
-rw-rw-r--      1       kw2022202065    kw2022202065    0       04 12 22:28     AA
-rw-rw-r--      1       kw2022202065    kw2022202065    72      04 11 20:06     Makefile
-rw-rw-r--      1       kw2022202065    kw2022202065    10      04 12 22:50     cle.out
-rwxrwxr-x      1       kw2022202065    kw2022202065    17200   04 12 21:25     cli
-rw-rw-r--      1       kw2022202065    kw2022202065    2240    04 12 23:09     cli.c
-rw-rw-r--      1       kw2022202065    kw2022202065    5       04 12 23:22     cli.out
-rwxrwxr-x      1       kw2022202065    kw2022202065    22312   04 12 23:08     srv
-rw-rw-r--      1       kw2022202065    kw2022202065    8344    04 12 23:10     srv.c
-rw-rw-r--      1       kw2022202065    kw2022202065    9       04 11 21:01     test
drwxrwxr-x      2       kw2022202065    kw2022202065    4096    04 12 23:19     testDir1/
drwxrwxr-x      2       kw2022202065    kw2022202065    4096    04 12 23:19     testDir2/
```

2) 추가 옵션을 입력하거나 존재하지 않는 경로를 입력하면 예외처리 된다.

```
kw2022202065@ubuntu:~/1-3$ ./cli dir -e | tee cli.out | ./srv
LIST -e
Error: invalid option
kw2022202065@ubuntu:~/1-3$ ./cli dir ./not_exist | tee cli.out | ./srv
LIST ./not_exist
`◆◆◆Ucannot access : No such directory
```

3. PWD (pwd)

pwd 를 하면 현재 경로가 출력되고, 추가 옵션이나 인자를 입력하면 예외처리 된다.

```
kw2022202065@ubuntu:~/1-3$ ./cli pwd | tee cli.out | ./srv
PWD
"/home/kw2022202065/1-3" is current directory
kw2022202065@ubuntu:~/1-3$ ./cli pwd -e | tee cli.out | ./srv
PWD -e
Error: invalid option
kw2022202065@ubuntu:~/1-3$ ./cli pwd arg | tee cli.out | ./srv
PWD arg
Error: argument is not required
```

4. CWD & CDUP (cd)

cd 에 추가 옵션을 넣으면 예외처리 된다. 경로를 입력하면 해당 경로로 이동한 뒤, 현재 디렉토리를 출력한다. 존재하지 않는 경로를 입력하면 예외처리 된다.

```
kw2022202065@ubuntu:~/1-3$ ./cli cd -e .. | tee cli.out | ./srv
CWD -e ..
Error: invalid option
kw2022202065@ubuntu:~/1-3$ ls
AA  cle.out  cli  cli.c  cli.out  Makefile  srv  srv.c  test  testDir1  testDir2
kw2022202065@ubuntu:~/1-3$ ./cli cd testDir1 | tee cli.out | ./srv
CWD testDir1
"/home/kw2022202065/1-3/testDir1" is current directory
kw2022202065@ubuntu:~/1-3$ ./cli cd .. | tee cli.out | ./srv
CDUP ..
```

```
kw2022202065@ubuntu:~/1-3$ ./cli cd ./not_exist | tee cli.out | ./srv
CWD ./not_exist
@◆◆◆◆UError: directory not found
```

5. MKD (mkdir)

mkdir 에 추가 옵션을 입력하거나, 이미 존재하는 디렉토리, 인자를 안 넣었을 경우에 예외처리 된다.

```
kw2022202065@ubuntu:~/1-3$ ./cli mkdir -e new_dir | tee cli.out | ./srv
MKD -e new_dir
Error: invalid option
kw2022202065@ubuntu:~/1-3$ ./cli mkdir new_dir1 | tee cli.out | ./srv
MKD new_dir1
kw2022202065@ubuntu:~/1-3$ ./cli mkdir new_dir2 new_dir3 | tee cli.out | ./srv
MKD new_dir2 new_dir3
kw2022202065@ubuntu:~/1-3$ ./cli mkdir new_dir1 | tee cli.out | ./srv
MKD new_dir1
Error: cannot create directory 'new_dir1': File exists
```

```
kw2022202065@ubuntu:~/1-3$ ./cli mkdir | tee cli.out | ./srv
Error: another argument is required
```

그 외에 정상적으로 입력된 것들에 대해서는 새로운 디렉토리가 생성된 것을 볼 수 있다.

```
kw2022202065@ubuntu:~/1-3$ ls
AA       cli    cli.out   new_dir1  new_dir3  srv.c  testDir1
cle.out  cli.c  Makefile  new_dir2  srv       test   testDir2
```

6. DELE (delete)

먼저 예제 파일을 생성한 뒤, delete 를 해보면 삭제된 것을 볼 수 있다. 다중 입력도 가능하다.

```
kw2022202065@ubuntu:~/1-3$ touch test1.dat
kw2022202065@ubuntu:~/1-3$ touch test2.dat
kw2022202065@ubuntu:~/1-3$ touch test3.dat
kw2022202065@ubuntu:~/1-3$ ls -al *.dat
-rw-rw-r-- 1 kw2022202065 kw2022202065 0 Apr 12 23:35 test1.dat
-rw-rw-r-- 1 kw2022202065 kw2022202065 0 Apr 12 23:35 test2.dat
-rw-rw-r-- 1 kw2022202065 kw2022202065 0 Apr 12 23:35 test3.dat
kw2022202065@ubuntu:~/1-3$ ./cli delete test1.dat | tee cli.out | ./srv
DELE test1.dat
kw2022202065@ubuntu:~/1-3$ ls -al test1.dat
ls: cannot access 'test1.dat': No such file or directory
kw2022202065@ubuntu:~/1-3$ ./cli delete test2.dat test3.dat | tee cli.out | ./srv
DELE test2.dat test3.dat
kw2022202065@ubuntu:~/1-3$ ls -al *.dat
ls: cannot access '*.dat': No such file or directory
```

추가 옵션을 넣으면 예외처리 된다.

```
kw2022202065@ubuntu:~/1-3$ touch test.dat
kw2022202065@ubuntu:~/1-3$ ./cli delete -e test.dat | tee cli.out | ./srv
DELE -e test.dat
`9s◆UError: invalid option
```

존재하는 파일과 존재하지 않는 파일을 동시에 입력한 경우, 존재하지 않는 파일에 대해서는 예외처리 되고 존재하는 파일은 정상적으로 삭제되는 것을 볼 수 있다.

```
kw2022202065@ubuntu:~/1-3$ ./cli delete not_exist test.dat | tee cli.out | ./srv
DELE not_exist test.dat
◆Error: failed to delete 'not_exist'
kw2022202065@ubuntu:~/1-3$ ls
AA       cli    cli.out   new_dir1  new_dir3  srv.c  testDir1
cle.out  cli.c  Makefile  new_dir2  srv       test   testDir2
```

7. RMD (rmdir)

존재하는 디렉토리를 입력할 시 정상적으로 삭제된다.

```
kw2022202065@ubuntu:~/1-3$ ls
AA       cli    cli.out   new_dir1  new_dir3  srv.c  testDir1
cle.out  cli.c  Makefile  new_dir2  srv       test   testDir2
kw2022202065@ubuntu:~/1-3$ ./cli rmdir new_dir1 | tee cli.out | ./srv
RMD new_dir1
kw2022202065@ubuntu:~/1-3$ ls -al new_dir1
ls: cannot access 'new_dir1': No such file or directory
```

```
kw2022202065@ubuntu:~/1-3$ ./cli rmdir new_dir2 new_dir3 | tee cli.out | ./srv
RMD new_dir2 new_dir3
```

이미 삭제된, 존재하지 않는 디렉토리를 입력할 시 예외처리 된다.

```
kw2022202065@ubuntu:~/1-3$ ./cli rmdir new_dir1 | tee cli.out | ./srv
RMD new_dir1
Error: falied to remove 'new_dir1'
```

```
kw2022202065@ubuntu:~/1-3$ ./cli rmdir not_exist | tee cli.out | ./srv
RMD not_exist
Error: falied to remove 'not_exist'
```

추가 인자를 입력하지 않았을 경우에도 예외처리 된다.

```
kw2022202065@ubuntu:~/1-3$ ./cli rmdir | tee cli.out | ./srv
Error: another argument is required
```

8. RNFR & RNTO (rename)

AA 를 bb 로 바꾼 결과이다.

```
kw2022202065@ubuntu:~/1-3$ ls
AA  cle.out  cli  cli.c  cli.out  Makefile  srv  srv.c  test  testDir1  testDir2
kw2022202065@ubuntu:~/1-3$ ./cli rename AA bb | tee cli.out | ./srv
RNFR AA
RNTO bb
@◆*◆,Vkw2022202065@ubuntu:~/1-3$ ls
bb  cle.out  cli  cli.c  cli.out  Makefile  srv  srv.c  test  testDir1  testDir2
```

추가 인자를 입력하지 않았거나, 이미 존재하는 파일 이름으로 바꾸려고 하는 경우 예외처리된다.

```
kw2022202065@ubuntu:~/1-3$ ./cli rename | tee cli.out | ./srv
Error: another argument is required
```

```
kw2022202065@ubuntu:~/1-3$ ./cli rename b test | tee cli.out | ./srv
RNFR b
RNTO test
PrD◆UError: name to change already exists
kw2022202065@ubuntu:~/1-3$ ls
b  cle.out  cli  cli.c  cli.out  Makefile  srv  srv.c  test  testDir1  testDir2
```

9. QUIT (quit)

정상적으로 입력한 경우 메시지 출력 뒤 종료된다. 추가 옵션이나 인자를 입력한 경우엔 예외처리 된다.

```
kw2022202065@ubuntu:~/1-3$ ./cli quit | tee cli.out | ./srv
QUIT
QUIT success
kw2022202065@ubuntu:~/1-3$ ./cli quit -e | tee cli.out | ./srv
QUIT -e
Error: invalid option
kw2022202065@ubuntu:~/1-3$ ./cli quit arg | tee cli.out | ./srv
QUIT arg
Error: argument is not required
```

고찰

이번 과제를 진행하면서 여러 시행착오를 겪었다. 다양한 예외처리를 하는 게 어려웠던 것 같다. 클라이언트에서 write 로 보내면, 서버가 read 로 변환된 명령어를 받는데 이를 다시 분리시키는 게 까다로웠다. 처음에 ls 부터 구현했는데, 옵션을 넣었을 때는 정상적으로 출력이 되었다가 옵션을 넣지 않고 단순히 ls 만 입력하는 현재 디렉토리 출력 명령어가 실행되지 않았다. 이에 대해 고민을 해보다가, 현재 디렉토리는 "."으로 저장되어 보내지는데 이때 read 로 받은 버퍼에 개행 문자가 문제임을 깨닫고 맨 처음에 버퍼의 개행 문자를 널문자로 고치는 작업을 추가하였다. 그러니까 잘 출력될 수 있었다. 하지만 이렇게 개행 문자를 없애버리니 후반에 rename 명령어에 대한 함수를 구현할 때 여러 오류가 났었다. rename 은 두 줄로 입력이 들어오는데, 개행 문자를 지워버리면서 두번째 라인 입력이 지워진 것이다. 그래서 처음에 버퍼를 save 버퍼로 따로 저장해놓은 뒤, rename 함수에서만 save 버퍼로 따로 수행하도록 변경하였다. 이 과정에서 몇몇 명령어들을 다중 인자로 실행했을 시 터미널에 다른 문자가 같이 출력되는 현상이 발생하였다. 버퍼관리 문제임은 알 수 있었지만, 이를 해결하는 게 쉽지 않았다. 따라서 아직까지 해당 문제는 완벽히 고치지는 못하였다. 버퍼관리에 대해서 조금 더 공부가 필요한 것 같다.

주로 ls 를 구현한 부분에서 파일 시스템에 대해 자세히 공부해 볼 수 있었던 것 같다. 특히 stat 정보들을 각각 불러오고 확인한 뒤 이를 출력하는 과정을 직접 구현해보면서, 파일 속성들을 더 잘 이해할 수 있었다. 그 외에 인자들을 분리하고 분석하는 과정이 어려웠지만 이번 과제를 통해 리눅스 명령어와 FTP 명령어들을 공부해볼 수 있었다.

Reference

강의자료 참고